

Polecenie: Dwumian Newtona. Napisać dwie wersje programu obliczającego symbol Newtona. W wersji iteracyjnej przechowywać trójkąt Pascala w zewnętrznej, statycznej tablicy kwadratowej.

Na początku zdefiniowałem podstawową bibliotekę <stdio.h> oraz funkcję o nazwie `int iteracyjnie(int n, int k)`

(Dodatkowo w funkcji za komentowany jest algorytm na liczenie dwumianu newtona przez iterację.)

Następnie określiłem tablicę kwadratową [100][100] ponieważ większe liczby i tak przekroczyłyby zakres int, oraz zdefiniowałem kilka zmiennych. Do n musiałem dodać 1 ponieważ z trójkąta Pascala całość jest przesunięta, a od k musiałem odjąć 1 z tego samego powodu.

Następnie algorytm trójkąta Pascala działa na zasadzie 2 pętli, jedna ogólna, druga szczegółowa.

Jeśli któraś ze zmiennych z pętli jest równa 0 to do tabeli [i][j] wpisywana jest wartość 1.

W przeciwnym wypadku do tabeli [i][j] program wpisuje wartość: `wynik = wynik * (i - j + 1) / j;`

Na końcu funkcja zwraca wartość tabeli o indeksach [n][k+1] tym samym zwracając wynik n nad k.

W dalszej części programu zdefiniowałem funkcję `int rekurencyjnie(int n, int k)` która liczy n nad k rekurencyjnie.

A więc jeśli k=n lub k=0 to funkcja zwróci wartość 1, w przeciwnym wypadku funkcja zwraca odwołanie do samej siebie `return rekurencyjnie(n - 1, k - 1) + rekurencyjnie(n - 1, k);`

lecz odpowiednio pomniejszoną. Dzięki temu funkcja liczy dwumian newtona.

Następnie wprowadziłem funkcję main() w której określiłem int nargs i char*argv[] będą one potrzebne do wyboru metody polecenia dwumianu newtona int main(int nargs, char *argv[]) Później określiłem zmienne pomocnicze oraz zmienną long long newton która zwróci wartość policzonego dwumianu newtona.

W dalszej części określiłem instrukcję warunkową switch dzięki której użytkownik może wybrać czy korzysta z metody 1 czy 2 (musi wpisać -M1 albo -M2).

Następnie program pobiera dane w postaci -Mx gdzie x to 1 albo 2 i jest to określenie metody oraz liczby z jakich ma być policzone k nad n.

Na koniec program wywołuje odpowiednią funkcję gdzie metoda 1 to dwumian newtona rekurencyjna, natomiast metoda 2 to dwumian newtona obliczony iteracyjnie i wypisuje wynik.

PRZYKŁAD DZIAŁANIA PROGRAMU:

REKURENCYJNIE i ITERACYJNIE:

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M1 8 4  
Dwumian newtona (z 8 nad 4) = 70 (użyta metoda: 1)
```

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M1 10 7  
Dwumian newtona (z 10 nad 7) = 120 (użyta metoda: 1)
```

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M1 6 6  
Dwumian newtona (z 6 nad 6) = 1 (użyta metoda: 1)
```

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M2 6 4  
Dwumian newtona (z 6 nad 4) = 15 (użyta metoda: 2)
```

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M1 2 8  
BLAD DANYCH!!! n musi byc wieksze od k  
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$
```